浙 江 林 学 院 学 报 2000, 17(3): 334~337
Journal of Zhejiang Forestry College

文章编号: 1000-5692(2000)03-0334-04

一类含2 个离步点的并行多步混合方法

李光辉, 李 洪
(浙江林学院 信息工程与基础科学系, 浙江 临安 311300)

摘要: 构造了一类能用3 个处理器并行实现的含2 个离步点的多步混合方法, 重点讨论了这类方法的稳定性、相容性及收敛性。参5

关键词: 刚性微分方程; 并行算法; 多步法; 混合法
中图分类号: TP301.6 **文献标识码**: A

刚性问题在自然界和工程技术领域常常遇到, 例如生态平衡问题、森林发育过程、化学反应过程、热核反应和飞行器运动等等。在用计算机求解时, 往往归结为大规模的刚性微分方程组。近 20 a 来, 该问题的并行算法研究随着超级并行计算机的发展, 更是引起了国内外众多学者的兴趣。

我们考虑 m 维的刚性系统

$$\begin{cases} y' = f(t, y(t)),\ t \in [a, b], \\ y(a) = \eta,\ \eta \in R^m。\end{cases} \tag{1}$$

这里 $m \geqslant 1, a < b, f: [a, b] \times R^m \to R^m$ 是给定充分光滑的映射, 且满足 Lipschitz 条件

$$\| f(t, y) - f(t, z) \| \leqslant L \| y - z \|,\ y, z \in R^m,$$

其中 $\|\circ\|$ 表示空间 R^m 中任意给定的一种范数, L 称为 Lipschitz 常数。

本文构造了求解问题(1)的一类含2个离步点的并行多步混合方法, 该类方法能用3个处理器并行实现, 并能有效提高计算精度。

1 方法的构造

我们构造求解问题(1)的如下形式的多步混合方法:

$$\begin{cases} y_{n+k} = \sum_{j=0}^{k-1} \alpha_j y_{n+j} + h\beta_k f(t_{n+k}, y_{n+k}) + h\beta_\nu f(t_{n+\nu}, y_{n+\nu}) + h\beta_\mu f(t_{n+\mu}, y_{n+\mu}), & \text{(2a)} \\ y_{n+\nu} = \sum_{j=-1}^{k-2} \alpha_j^{(1)} y_{n+j} + h\beta_\nu^* f(t_{n+\nu}, y_{n+\nu}), & \text{(2b)} \\ y_{n+\mu} = \sum_{j=-1}^{k-2} \alpha_j^{(2)} y_{n+j} + h\beta_\mu^* f(t_{n+\mu}, y_{n+\mu})。 & \text{(2c)} \end{cases}$$

这里恒设 $\beta_k, \beta_\nu, \beta_\mu, \beta_\nu^* \neq 0$, 且 $\nu, \mu \neq 0, 1, 2, \cdots\cdots, k$。显然, 方法(2)中含2个离步点信息 $y_{n+\nu}$ 与 $y_{\mu+\mu}$。若令:

$$Y^{(n)} = [y_{n+k}^T, y_{n+\nu}^T, y_{n+\mu}^T]^T,$$
$$y^{(n)} = [y_n^T, y_{n+1}^T, \cdots, y_{n+k}^T]^T,$$

收稿日期: 1999-11-15; 修回日期: 2000-03-27

作者简介: 李光辉(1970—), 男, 湖南资兴人, 讲师, 硕士, 主要从事数值并行算法及计算机图形学研究。

$$\xi_n = y_{n+k},\quad \beta = [0, \cdots, 0, 1] \in R^{k+1},$$
$$F(t_n, Y^{(n)}) = [f(t_{n+k}, y_{n+k})^T, f(t_{n+\nu}, y_{n+\nu})^T, f(t_{n+\mu}, y_{n+\mu})^T]^T,$$

那么方法(2)可写成一般线性方法的形式：

$$\begin{cases} Y^{(n)} = hC_{11}F(t_n, Y^{(n)}) + C_{12}y^{(n-1)}, \\ y^{(n)} = hC_{21}F(t_n, Y^{(n)}) + C_{22}y^{(n-1)}, \\ \xi_n = \beta y^{(n)}。 \end{cases} \tag{3}$$

其中：

$$C_{11} = \begin{bmatrix} \beta_k & \beta_\nu & \beta_\mu \\ 0 & \beta_\nu^* & 0 \\ 0 & 0 & \beta_\mu^* \end{bmatrix},\quad C_{12} = \begin{bmatrix} 0 & \alpha_0 & \alpha_1 & \cdots & \alpha_{k-2} & \alpha_{k-1} \\ \alpha_{-1}^{(1)} & \alpha_0^{(1)} & \alpha_1^{(1)} & \cdots & \alpha_{k-2}^{(1)} & 0 \\ \alpha_{-1}^{(2)} & \alpha_0^{(2)} & \alpha_1^{(2)} & \cdots & \alpha_{k-2}^{(2)} & 0 \end{bmatrix},$$

$$C_{21} = \begin{bmatrix} 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ \beta_k & \beta_\nu & \beta_\mu \end{bmatrix} \in R^{(k+1)\times 3},\quad C_{22} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & \alpha_0 & \alpha_1 & \cdots & \alpha_{k-2} & \alpha_{k-1} \end{bmatrix} \in R^{(k+1)\times(k+1)}。$$

2　方法的稳定性、相容性及收敛性

为了讨论方便，我们首先引入 2 个重要概念：对角稳定和级阶。

定义 1

称方法（3）是对角稳定的，若存在对角矩阵 $D > 0$，使得

$$C_{11}^T D + DC_{11} > 0。$$

定义 2

称方法（3）的级阶为 p，如果 p 是具有下面性质的最大非负整数：对任给的初值问题（1）及任给的步长 $h \in (0, h_1)$，有以下不等式成立：

$$\| \Delta^h(t) \| \leqslant d_1 h^{p+1},\quad \| \delta^h(t) \| \leqslant d_2 h^{p+1},\quad \| \sigma^h(t) \| \leqslant d_3 h^p。$$

这里 $h_1 > 0$，且要求当 $h \in (0, h_1)$ 时，所有涉及的时间节点均不越出积分区间 $[a, b]$，每个 $d_i (i = 1, 2, 3)$ 仅依赖于方法及某些导数界 M_i；$\Delta^h(t)$，$\delta^h(t)$，$\sigma^h(t)$ 由下面方程确定：

$$\begin{cases} Y(t) = hC_{11}F(t, Y(t)) + C_{12}H(t-h) + \Delta^h(t), \\ H(t) = hC_{21}F(t, Y(t)) + C_{22}H(t-h) + \delta^h(t), \\ y(t+kh) = \beta H(t) + \sigma^h(t)。 \end{cases} \tag{4}$$

定理 1

当 $\beta_k, \beta_\nu^*, \beta_\mu^* > 0$ 时，方法（3）是对角稳定的。

证明

只须取常数 d_1, d_2 满足

$$\begin{cases} 4d_1\beta_k\beta_\nu^* > \beta_\nu^2, \\ d_2\beta_\mu^*(4d_1\beta_k\beta_\nu^* - \beta_\nu^2) > d_1\beta_\nu^*\beta_\mu^2, \end{cases}$$

令 $D = \mathrm{diag}(1, d_1, d_2) > 0$ 时，因 $\beta_k, \beta_\nu^*, \beta_\mu^* > 0$，矩阵

$$DC_{11} + C_{11}^T D = \begin{bmatrix} 2\beta_k & \beta_\nu & \beta_\mu \\ \beta_\nu & 2d_1\beta_\nu^* & 0 \\ \beta_\mu & 0 & 2d_2\beta_\mu^* \end{bmatrix},$$

显然是正定的，故方法（3）对角稳定。

定理 2

当方法（3）的系数满足条件

$$\begin{cases} \sum_{j=0}^{k-1}\alpha_j = 1,\ \sum_{j=-1}^{k-2}\alpha_j^{(1)} = 1,\ \sum_{j=-1}^{k-2}\alpha_j^{(2)} = 1, \\ \sum_{j=0}^{k-1} j^p\alpha_j + pk^{p-1}\beta_k + p\nu^{p-1}\beta_\nu + p\mu^{p-1}\beta_\mu = k^p,\quad p = 1, 2, \cdots, k, \\ \sum_{j=-1}^{k-2} j^p\alpha_j^{(1)} + p\nu^{p-1}\beta_\nu^* = \nu^p, p = 1, 2, \cdots, k, \\ \sum_{j=-1}^{k-2} j^p\alpha_j^{(2)} + p\mu^{p-1}\beta_\mu^* = \mu^p,\quad p = 1, 2, \cdots, k \end{cases} \tag{5}$$

时，该方法的级阶为 k。

证明

将 $Y(t_n)=[y(t_n+kh)^T, y(t_n+\nu h)^T, y(t_n+\mu h)^T]^T$，$H(t_n)=[y(t_n)T, y(t_n+h)^T, \cdots, y(t_n+kh)^T]^T$ 分别代入（4），并作 Taylor 展开，可算出

$$\begin{cases} \Delta_1^h(t_n) = O(h^{k+1}),\quad \Delta_2^h(t_n) = O(h^{k+1}),\quad \Delta_3^h(t_n) = O(h^{k+1}), \\ \delta_i^h(t_n) = 0,\quad i = 1, 2, \cdots, k,\quad \delta_{k+1}^h(t_n) = \Delta_1^h(t_n) = O(h^{k+1}), \\ \sigma^h(t_n) = 0. \end{cases}$$

这里，$\Delta_i^h(t_n)(i=1, 2, 3)$ 表示 $\Delta^h(t_n)$ 的各个分量，$\delta_i^h(t_n)(i=1, 2, \cdots, k+1)$ 则表示 $\delta^h(t_n)$ 的各个分量。因此有

$$\|\Delta^h(t_n)\| \leqslant d_1 h^{k+1},\quad \|\delta^h(t_n)\| \leqslant d_2 h^{k+1},\quad \|\sigma^h(t_n)\| = 0,$$

其中：d_1, d_2 仅依赖于方法及其真解 $y(t)$ 的某些导数界，从而推出方法（3）的级阶为 k。

在下文中，我们用方法（3）～（5）表示满足条件（5）的方法（3）。

推论 1

方法（3）～（5）是 k 阶 B－相容的。

推论 2

如果多项式 $\det(\lambda I - C_{22}) = \lambda(\lambda^k - \alpha_{k-1}\lambda^{k-1} - \cdots - \alpha_1\lambda - \alpha_0)$ 满足根条件（即所有特征根的模不超过 1，模为 1 的特征根为单根），那么方法（3）～（5）是 k 阶经典收敛的。

以上推论的证明及有关概念可参见文献[2]。

3 方法的实现

对任意给定的 $\nu, \beta_\nu, \mu, \beta_\mu$，容易证明，只要

$$\det\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 & 1 \\ -1 & 1 & 2 & \cdots & k-2 & 2\nu \\ (-1)^2 & 1^2 & 2^2 & \cdots & (k-2)^2 & 3\nu^2 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ (-1)^{k-1} & 1^{k-1} & 2^{k-1} & k\nu^{k-1} & (k-2)^{k-1} & k\nu^{k-1} \end{bmatrix} \neq 0,$$

就可从方程（5）中唯一解出方法（2）的其余系数。因此方法（2）～（5）可以看成一族含有 4 个自由参数 $\nu, \beta_\nu, \mu, \beta_\mu$ 的 $k+1$ 步 k 阶混合方法（这里方法（2）与方法（3）是等价的）。该方法的级阶为 k，这对提高数值求解刚性问题的精度十分有利。

该算法族的另一优点是可用 3 个处理器并行实现。事实上，当已获得背后值 $y_n, y_{n+1}, \cdots, y_{n+k-1}$ 及 2 个离步点信息 $y_{n+\nu}$ 与 $y_{n+\mu}$ 时，可用 1 个处理器通过方程（2a）计算 y_{n+k}，同时用另 2 个处理器分别通过方程（2b）和（2c）右移一步计算 $y_{n+\nu+1}$ 与 $y_{n+\mu+1}$。3 个并行的计算进程如下：

$$
\begin{cases}
y_{n+k} = \sum_{j=0}^{k-1} \alpha_j y_{n+j} + h\beta_k f(t_{n+k}, y_{n+k}) + h\beta_\nu f(t_{n+\nu}, y_{n+\nu}) + h\beta_\mu f(t_{n+\mu}, y_{n+\mu}), \\
y_{n+\nu+1} = \sum_{j=0}^{k-1} \alpha_{j-1}^{(1)} y_{n+j} + h\beta_\nu^* f(t_{n+\nu+1}, y_{n+\nu+1}), \\
y_{n+\mu+1} = \sum_{j=0}^{k-1} \alpha_{j-1}^{(2)} y_{n+j} + h\beta_\mu^* f(t_{n+\mu+1}, y_{n+\mu+1})。
\end{cases}
$$

显然，这里每个处理器仅须求解 1 个含 m 个未知数的非线性方程组，这和用 BDF 方法进行求解时每一积分步的计算量是一样的。因此可以预期，在并行环境下使用方法（2）～（5）将具有和使用 BDF 方法大体上相同的计算速度。为了获得稳定性尽可能好的方法类，对于自由参数 ν, β_ν, μ, β_μ，我们可以运用优化软件，通过计算机全面搜索。具体衡量指标是方法稳定域的大小，稳定程度和 stiff 参数等。此外，还应该通过数值试验来检验所获得的方法类。

参考文献：

[1] 李光辉，李寿佛．修改的并行多步混合方法[J]．湘潭大学自然科学学报，1999，**21**（2）：20—40.

[2] 李寿佛．刚性微分方程算法理论[M]．长沙：湖南科学技术出版社，1997，248—254.

[3] Lambert J. D. *Computational methods in ordinary differential equations*[M]. New Jork：John Wiley &Sons，1973. 162—185.

[4] 刘德贵，费景高，韩天敏．刚性大系统数字仿真方法[M]．郑州：河南科学技术出版社，1996. 160—164.

[5] Butcher J C. Linear and nonlinear stability for general linear methods[J]. *BIT*，1987，**27**（2）：182—189.

A class of parallel multistep hybrid methods with two off-step points

LI Guang-hui，LI Hong

（Department of Information Engineering and Basic Science，Zhejiang Forestry College，Lin'an 311300，Zhejiang，China）

Abstract：A class of multistep hybrid methods is constructed. These methods can be carried out by three processors. The stability，consistency and convergence properities of these methods are discussed.

Key words：stiff differential equations；parallel algorithms；multistep methods；hybrid methods